

Amendment to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A classfile modification method, comprising:

~~converting said classfile into a collection of objects whose organization is derived from said classfile's organization, said collection of objects comprising unique objects for each method information structure found in said classfile, each of said unique objects referenced to other objects in said organization, said other objects representing their corresponding unique object's method information structure's byte code instructions;~~

~~adding at least one additional other object to an arrangement of other objects that are referenced to a unique object, said at least one additional other object corresponding to at least one byte code instruction that causes a plug-in module's handler method to provide output function treatment for said unique object's method; and,~~

~~converting the resulting collection of objects into a modified version of said classfile.~~

converting a classfile into a collection of objects, said collection of objects including a first object that represents a method information structure found in said classfile, said collection of objections including a second object that inherits a property that said first object has, said second object representing a bytecode instruction of a method;

adding a third object to said collection of objects, said third object inheriting the properties of said first object, said third object representing a bytecode instruction to be executed by said method that, when executed by said method, invokes a pre-existing

dispatcher to identify a plug-in module for said method that said method invokes to report and/or record information about said method;

adding a fourth object to said collection of objects that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a second method that registers, with said dispatcher upon loading of said classfile, an identity of said classfile's class and respective identities of methods of said classfile, said dispatcher and plug-in module being in existence prior to said loading of said classfile; and,

converting said collection of objects including said third object into a modified version of said classfile.

2-4. (Canceled).

5. (Currently Amended) The classfile modification method of claim 4 1, wherein ~~said modifying further comprising adding comprises creating a new fifth object to said collection of objects~~ that represents a new field information structure for said classfile.

6. (Currently Amended) The classfile modification method of claim 5, wherein said field information structure is to store a numeric identification assigned to said class by said dispatcher.

7. (Canceled).

8. (Currently Amended) The classfile modification method of claim 1, wherein ~~said adding at least one additional other object further comprises adding an additional other third object at~~ is added to said collection of objects in a position that corresponds

to a region of said ~~unique~~ method's instructions that is executed just after said ~~unique~~ method's entry point is reached.

9. (Currently Amended) The classfile modification method of claim 8₁ wherein said classfile is a Java compatible classfile and said ~~additional-other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

10. (Currently Amended) The classfile modification method of claim 8₁ wherein said classfile is a Java compatible classfile and said ~~additional-other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

11. (Currently Amended) The classfile modification method of claim 1₁ wherein said ~~adding at least one additional-other object further comprises adding an additional other~~ third object ~~at~~ is added to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed if an exit point of said ~~unique~~ method will inevitably be reached.

12. (Currently Amended) The classfile modification method of claim 11₁ wherein said classfile is a Java compatible classfile and said ~~additional-other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

13. (Currently Amended) The classfile modification method of claim 11₁ wherein said classfile is a Java compatible classfile and said ~~additional-other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

14. (Currently Amended) The classfile modification method of claim 1₁ wherein said ~~adding at least one additional-other object further comprises adding an additional~~

~~other~~ third object ~~at is added to said collection of objects in~~ a position that corresponds to a region of said ~~unique~~ method's instructions that will be executed if an error arises during execution of said ~~unique~~ method.

15. (Currently Amended) The classfile modification method of claim 14, wherein said classfile is a Java compatible classfile and said ~~additional-other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

16. (Currently Amended) The classfile modification method of claim 14, wherein said classfile is a Java compatible classfile and said ~~additional-other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

17. (Currently Amended) The classfile modification method of claim 1, further comprising ~~wherein said adding at least one additional-other object further comprises:~~

adding a ~~first additional-other~~ said third object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed just after said ~~unique~~ method's entry point is reached;

adding a ~~second additional-other~~ fifth object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed if an exit point of said ~~unique~~ method will inevitably be reached; and,

adding a ~~third additional~~ sixth object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that will be executed if an error arises during execution of said ~~unique~~ method.

18. (Currently Amended) The classfile modification method of claim 17, wherein said classfile is a Java compatible classfile and said ~~first, second and third additional~~

~~other third, fifth and sixth~~ objects correspond to the addition of invokestatic instructions.

19. (Currently Amended) The classfile modification method of claim 17, wherein said classfile is a Java compatible classfile and said ~~first, second and third~~ additional ~~other third, fifth and sixth~~ objects correspond to the addition of invokevirtual instructions.

20. (Currently Amended) A machine readable medium containing instructions which when executed cause a classfile modification method to be performed, said classfile modification method comprising:

~~converting said classfile into a collection of objects whose organization is derived from said classfile's organization, said collection of objects comprising unique objects for each method information structure found in said classfile, each of said unique objects referenced to other objects in said organization, said other objects representing their corresponding unique object's method information structure's byte code instructions;~~

~~adding at least one additional other object to an arrangement of other objects that are referenced to a unique object, said at least one additional other object corresponding to at least one byte code instruction that causes a plug-in module's handler method to provide output function treatment for said unique object's method; and,~~

~~converting the resulting collection of objects into a modified version of said classfile.~~

converting a classfile into a collection of objects, said collection of objects including a first object that represents a method information structure found in said classfile, said collection of objections including a second object that inherits a property

that said first object has, said second object representing a bytecode instruction of a method;

adding a third object to said collection of objects, said third object inheriting the properties of said first object, said third object representing a bytecode instruction to be executed by said method that, when executed by said method, invokes a pre-existing dispatcher to identify a plug-in module for said method that said method invokes to report and/or record information about said method;

adding a fourth object to said collection of objects that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a second method that registers, with said dispatcher upon loading of said classfile, an identity of said classfile's class and respective identities of methods of said classfile, said dispatcher and plug-in module being in existence prior to said loading of said classfile; and,

converting said collection of objects including said third object into a modified version of said classfile.

21-23. (Canceled).

24. (Currently Amended) The machine readable medium of claim 23, 20, ~~wherein said modifying~~ wherein said classfile modification method further comprises adding comprises creating a new fifth object to said collection of objects that represents a new field information structure for said classfile.

25. (Currently Amended) The machine readable medium of claim 24, wherein said field information structure is to store a numeric identification assigned to said class by said dispatcher.

26. (Canceled).

27. (Currently Amended) The machine readable medium of claim 20₁ wherein said ~~adding at least one additional other object further comprises adding an additional other~~ third object at is added to said collection of objects in a position that corresponds to a region of said ~~unique~~ method's instructions that is executed just after said ~~unique~~ method's entry point is reached.

28. (Currently Amended) The machine readable medium of claim 27₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

29. (Currently Amended) The machine readable medium of claim 27₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

30. (Currently Amended) The machine readable medium of claim 20₁ wherein said ~~adding at least one additional other object further comprises adding an additional other~~ third object at is added to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed if an exit point of said ~~unique~~ method will inevitably be reached.

31. (Currently Amended) The machine readable medium of claim 30₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

32. (Currently Amended) The machine readable medium of claim 30₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

33. (Currently Amended) The machine readable medium of claim 20₁ wherein said ~~adding at least one additional other object further comprises adding an additional other~~ third object ~~at~~ is added to said collection of objects in a position that corresponds to a region of said ~~unique~~ method's instructions that will be executed if an error arises during execution of said ~~unique~~ method.

34. (Currently Amended) The machine readable medium of claim 33₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

35. (Currently Amended) The machine readable medium of claim 33₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

36. (Currently Amended) The machine readable medium of claim 20₁ wherein said classfile modification method further comprises ~~wherein said adding at least one additional other object further comprises:~~

~~adding a first additional other~~ said third object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed just after said ~~unique~~ method's entry point is reached;

~~adding a second additional other~~ fifth object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed if an exit point of said ~~unique~~ method will inevitably be reached; and,

adding a ~~third additional~~ sixth object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that will be executed if an error arises during execution of said ~~unique~~ method.

37. (Currently Amended) The machine readable medium of claim 36₁, wherein said classfile is a Java compatible classfile and said ~~first, second and third additional~~ ~~other~~ third, fifth and sixth objects correspond to the addition of ~~invokestatic~~ instructions.

38. (Currently Amended) The machine readable medium of claim 36₁, wherein said classfile is a Java compatible classfile and said ~~first, second and third additional~~ ~~other~~ third, fifth and sixth objects correspond to the addition of ~~invokevirtual~~ instructions.

39. (Currently Amended) A computing system implemented with a machine readable medium containing instructions which when executed cause a classfile modification method to be performed, said classfile modification method comprising:

~~converting said classfile into a collection of objects whose organization is derived from said classfile's organization, said collection of objects comprising unique objects for each method information structure found in said classfile, each of said unique objects referenced to other objects in said organization, said other objects representing their corresponding unique object's method information structure's byte code instructions;~~

~~adding at least one additional other object to an arrangement of other objects that are referenced to a unique object, said at least one additional other object corresponding to at least one byte code instruction~~

~~that causes a plug-in module's handler method to provide output function treatment for said unique object's method; and,~~

~~converting the resulting collection of objects into a modified version of said classfile.~~

converting a classfile into a collection of objects, said collection of objects including a first object that represents a method information structure found in said classfile, said collection of objections including a second object that inherits a property that said first object has, said second object representing a bytecode instruction of a method;

adding a third object to said collection of objects, said third object inheriting the properties of said first object, said third object representing a bytecode instruction to be executed by said method that, when executed by said method, invokes a pre-existing dispatcher to identify a plug-in module for said method that said method invokes to report and/or record information about said method;

adding a fourth object to said collection of objects that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a second method that registers, with said dispatcher upon loading of said classfile, an identity of said classfile's class and respective identities of methods of said classfile, said dispatcher and plug-in module being in existence prior to said loading of said classfile; and,

converting said collection of objects including said third object into a modified version of said classfile.

40-42. (Canceled).

43. (Currently Amended) The machine readable medium of claim 42, ~~wherein said modifying~~ wherein said classfile modification method further comprises adding

~~comprises creating a new fifth object to said collection of objects~~ that represents a new field information structure for said classfile.

44. (Currently Amended) The machine readable medium of claim 43₁ wherein said field information structure is to store a numeric identification assigned to said class by said dispatcher.

45. (Canceled).

46. (Currently Amended) The machine readable medium of claim 39₁ wherein ~~said adding at least one additional other object further comprises adding an additional other third object at~~ is added to said collection of objects in a position that corresponds to a region of said ~~unique~~ method's instructions that is executed just after said ~~unique~~ method's entry point is reached.

47. (Currently Amended) The machine readable medium of claim 46₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

48. (Currently Amended) The machine readable medium of claim 46₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

49. (Original) The machine readable medium of claim 39 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached.

50. (Currently Amended) The machine readable medium of claim 49₁ wherein said ~~adding at least one additional other object further comprises adding an additional other~~ third object at is added to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed if an exit point of said ~~unique~~ method will inevitably be reached.

51. (Currently Amended) The machine readable medium of claim 49₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

52. (Currently Amended) The machine readable medium of claim 39₁ wherein said ~~adding at least one additional other object further comprises adding an additional other~~ third object at is added to said collection of objects in a position that corresponds to a region of said ~~unique~~ method's instructions that will be executed if an error arises during execution of said ~~unique~~ method.

53. (Currently Amended) The machine readable medium of claim 52₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokestatic instruction.

54. (Currently Amended) The machine readable medium of claim 52₁ wherein said classfile is a Java compatible classfile and said ~~additional other~~ adding of said third object corresponds to the addition of an invokevirtual instruction.

55. (Currently Amended) The machine readable medium of claim 39₁ wherein said classfile modification method further comprises ~~wherein said adding at least one additional other object further comprises:~~

adding a ~~first additional other~~ said third object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed just after said ~~unique~~ method's entry point is reached;

adding a ~~second additional other~~ fifth object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that is executed if an exit point of said ~~unique~~ method will inevitably be reached; and,

adding a ~~third additional~~ sixth object to said collection of objects at a position that corresponds to a region of said ~~unique~~ method's instructions that will be executed if an error arises during execution of said ~~unique~~ method.

56. (Currently Amended) The machine readable medium of claim 55, wherein said classfile is a Java compatible classfile and said ~~first, second and third additional other~~ third, fifth and sixth objects correspond to the addition of invokestatic instructions.

57. (Currently Amended) The machine readable medium of claim 55, wherein said classfile is a Java compatible classfile and said ~~first, second and third additional other~~ third, fifth and sixth objects correspond to the addition of invokevirtual instructions.